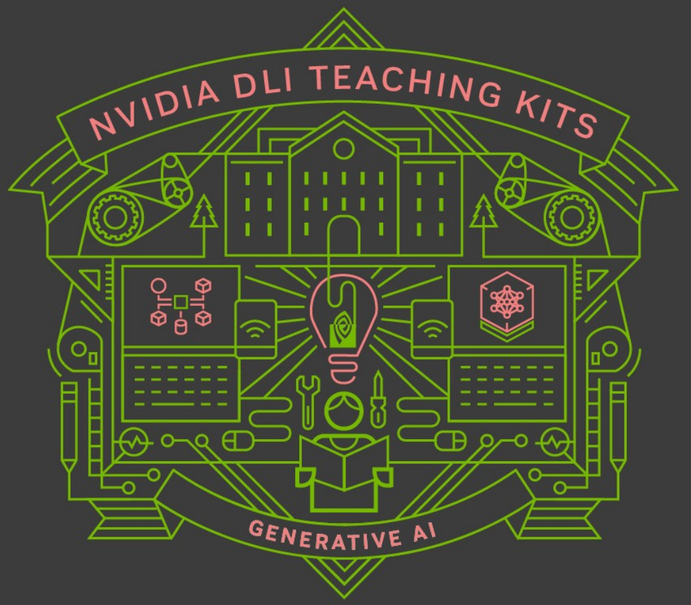
Lecture 2.1 - Natural Language Processing and Language Modeling

Generative AI Teaching Kit



This lecture

- Introduction to NLP: History and Applications
- Classical NLP Methods
- Language Models: From Basics to Modern Approaches
- Transformers and the Rise of Modern LMs

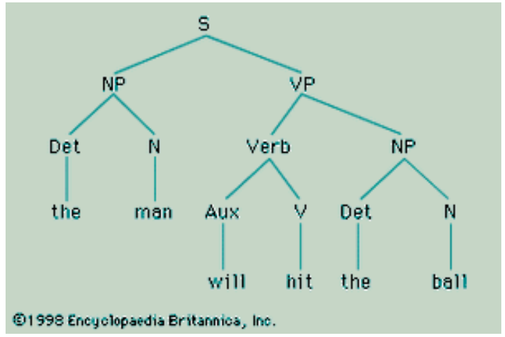
Introduction to NLP: History and Applications

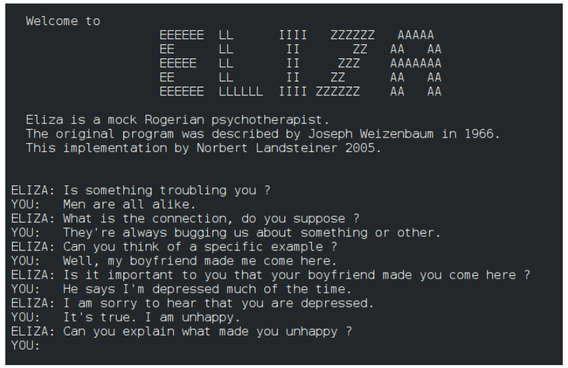
Introduction to NLP: History and Applications

1950s – 1960s: Rules-based

Challenges:

- Scalability issues due to manual rule creation.
- Limited ability to handle ambiguity in natural language.

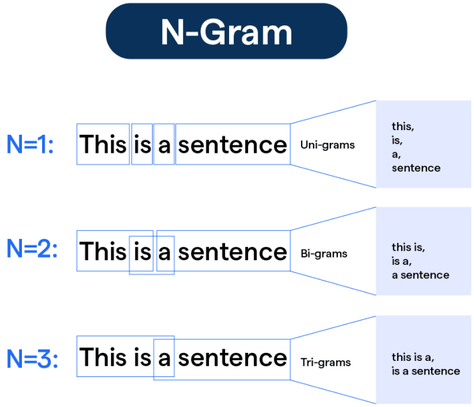




1960s – 1980s: Statistical Methods

Challenges:

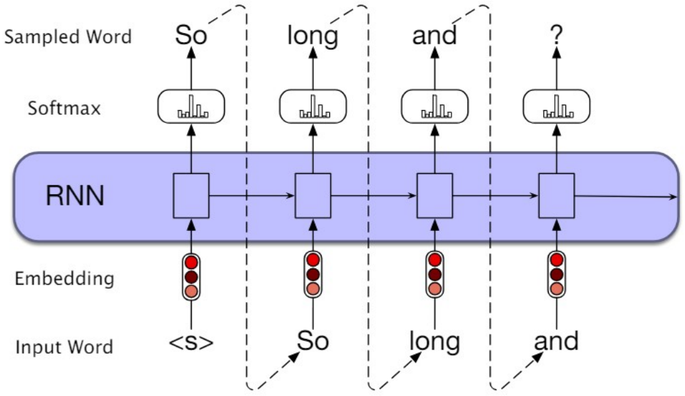
- Dependence on large labeled datasets.
- Difficulty handling long-range dependencies in language.



1990s – 2010s: Machine Learning Methods

Challenges:

- Difficulty scaling RNNs to handle very long sequences.
- Training inefficiencies for deep networks.



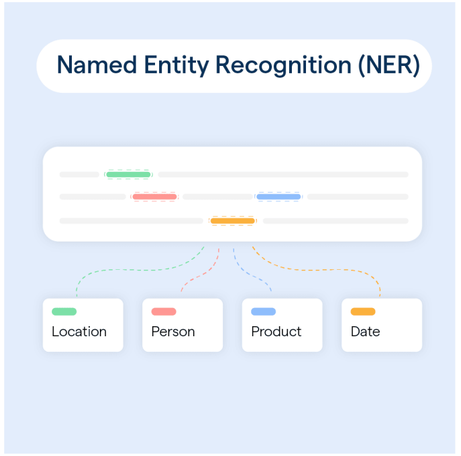
Introduction to NLP: History and Applications

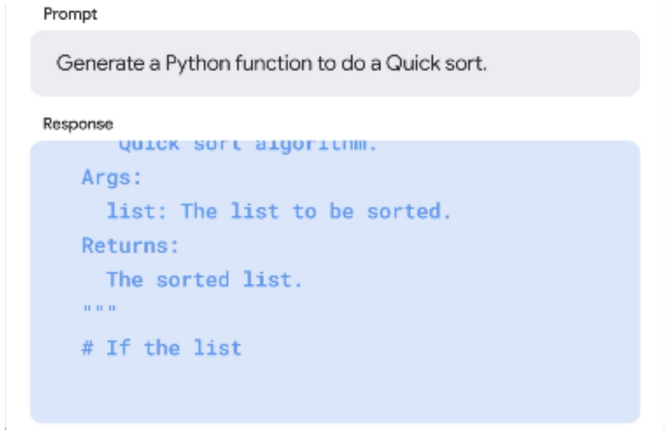
NLP is used for a wide variety of applications:

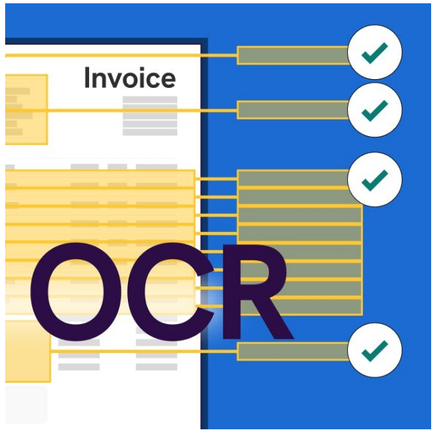
Classification
- Sentiment Analysis
- Text Classification
- Named Entity Recognition (NER)
- Semantic Analysis
- Grammar and Spell Checking

Generation
- Language Translation
- Chatbots
- Text Summarization
- Code Generation

Multimodal
- Text-to-Image Descriptions
- Text-to-Speech (TTS)
- Optical Character Recognition (OCR)







Classical NLP Methods

Classical NLP Methods

Prior to modern deep learning-based NLP approaches, other approaches like rules-based and statistics-based models were popular:

Rules-based

Rule-based models use a set of predefined rules to interpret and understand natural language. These models work by breaking down text into smaller parts, such as words or phrases, and then applying a set of rules to those parts to extract meaning. The rules can be simple or complex and are usually created the humans to understand the language and perform tasks.

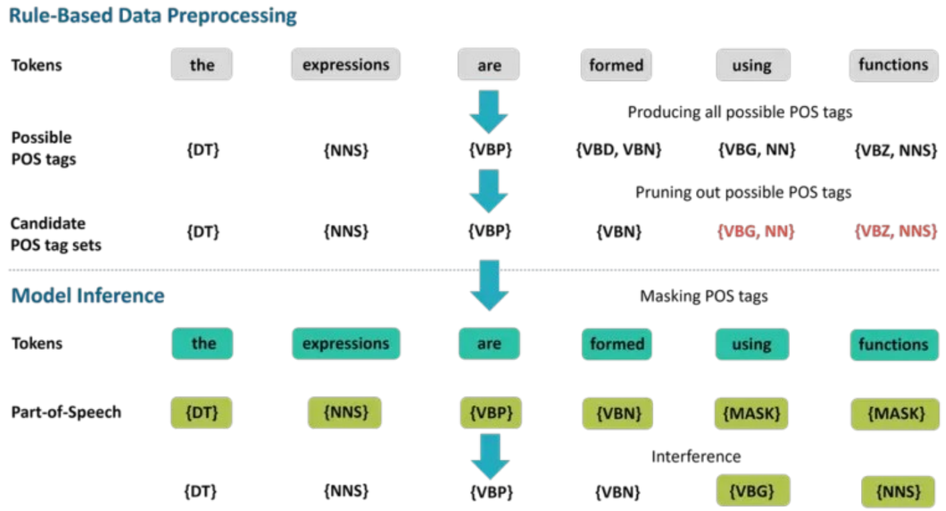
Statistics-based

- N-gram Model

N-gram word models are based on an assumption that the probability of the next word in a sequence depends only on a fixed size window of previous words. If only one previous word is considered, it is called a bigram model; if two words, a trigram model; if n − 1 words, an n-gram model.

- Hidden Markov Model

A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. In a regular Markov model (Markov Model), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible.



Example:

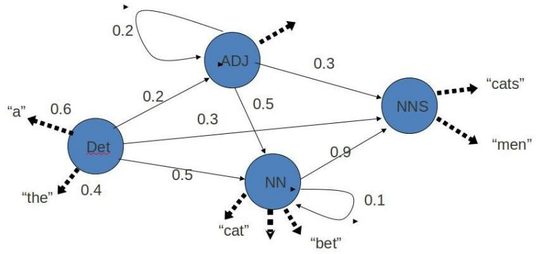
Compute the probability of the next word for

"This is the" is "house".

Solution: $\frac{Count(\text{"this is the house"})}{Count(\text{"this is the"})} = 0.25.$

Training Corpus

This is the house that Jack built.
This is the malt
That lay in the house that Jack built.
This is the rat,
That ate the malt,
That lay in the house that Jack built.
This is the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.



DARTMOUTH ENGINEERING | NVIDIA

Classical NLP Methods - Limitations

1. Lexical Ambiguity
Words with multiple meanings (e.g., “bank” as a financial institution vs. river bank).

- **Syntactic Ambiguity:**

Different ways to parse a sentence (e.g., “I saw the man with a telescope”).

- **Contextual Limitations:**

Classical methods often struggle with determining meaning based on context.

2. Fixed-Size Context Windows:
N-gram models and early statistical methods consider only short-term dependencies.

Classical NLP Methods - Limitations

3. Handling Out-of-Vocabulary (OOV) Words

- **Vocabulary Limitations:**

Classical methods struggle with unseen words or variations (e.g., “tweeted” vs. “tweet”).

- **Morphological Variations:**

Inflexible handling of word forms (e.g., “run,” “running,” “ran”).

4. Difficulty in Capturing Semantics

- **Surface-Level Understanding:**

Classical methods focus on syntax and word frequency rather than meaning.

- **No Contextual Word Representation:**

Words are treated independently, missing nuances and context (e.g., “bat” as an animal vs. sports equipment).

Classical NLP Methods - Limitations

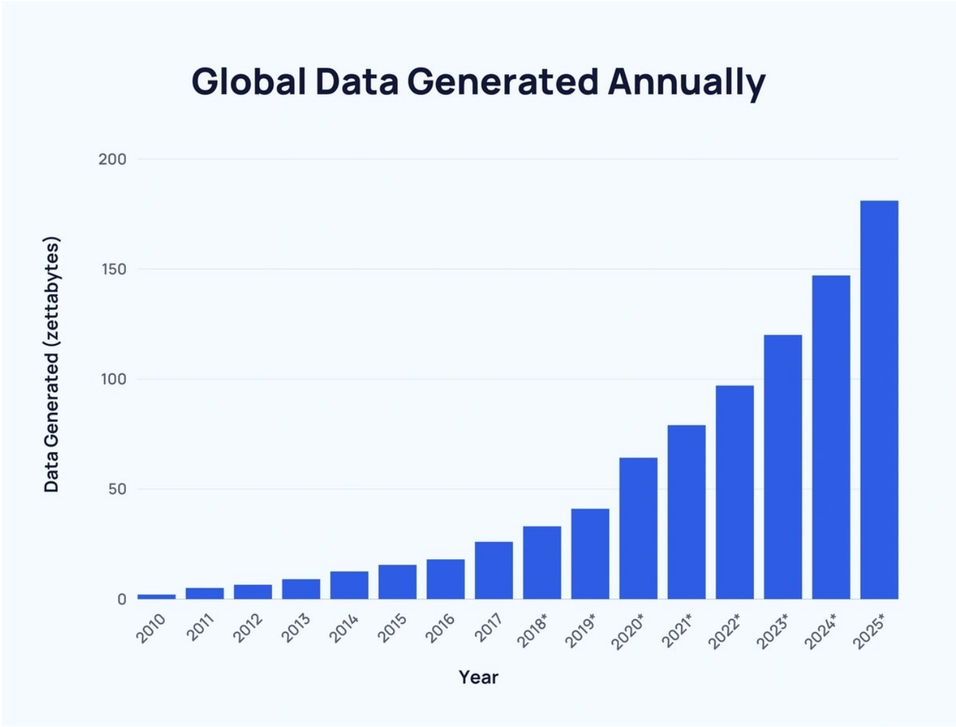
5. Poor Scalability with Large Data

- **Rule Explosion:**

As data size grows, maintaining and expanding rule sets becomes impractical.

- **Statistical Models:**

Often require extensive feature engineering and large labeled datasets.



Language Models: From Basics to Modern Approaches

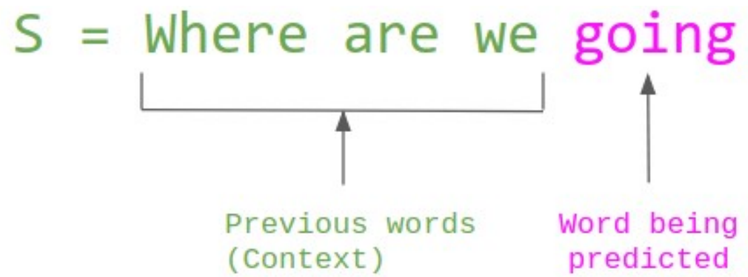
Language Models: From Basics to Modern Approaches

What are language models/language modeling?

Language modeling is the process of assigning probabilities to sequences of words, capturing how likely a sequence is to occur.

$$P_{(w_1,w_2,\ldots,w_n)} = p(w_1)p(w_2|w_1)p(w_3|w_1,w_2)...p(w_n|w_1,w_2,..,w_{n-1})$$
$$= \prod_{i=1}^{n} p(w_i|w_1,...,w_{i-1})$$

$p(w_2|w_1)$ - is the probability of word w_2 given that word w_1 is the previous word. Typically, the word with the highest probability in the vocabulary is then selected as the next word.



P(S) = P(Where) x P(are | Where) x P(we | Where are) x P(going | Where are we)

Language Models: From Basics to Modern Approaches

Bag of Words (BoW)

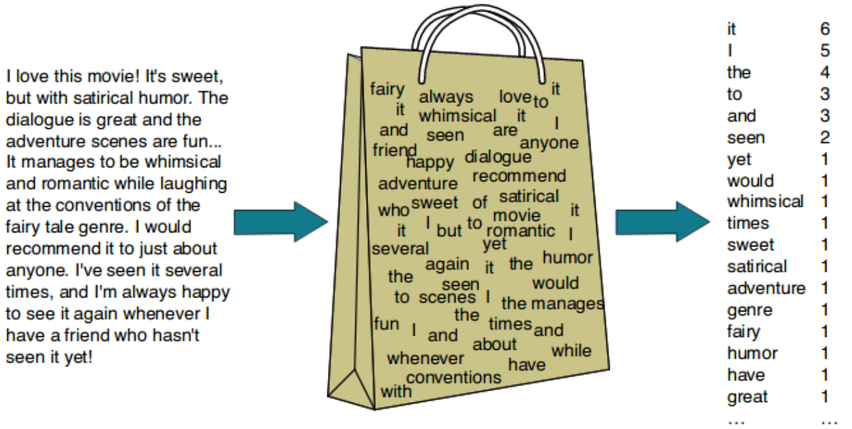
Description:
Represents text as an unordered collection of words, ignoring grammar and word order.

Pros:

- Simple and Efficient: Easy to implement and computationally inexpensive.
- Good for Short Texts: Works well when context and order aren't crucial (e.g., spam detection).

Cons:

- Ignores Word Order: Loses context and meaning (e.g., "not good" vs. "good").
- Sparse Representation: Large vocabulary leads to high-dimensional, sparse vectors.
- Contextual Limitations: Struggles with understanding relationships between words.



N-Gram Models

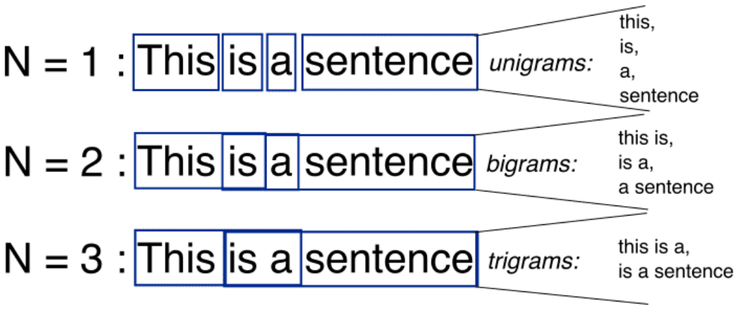
Description:
Represents text as sequences of [OBJ]-length contiguous words (e.g., bigrams = 2-word sequences).

Pros:

- Captures Word Order: Preserves local context and basic syntax.
- Better Context Representation: Improves performance over BoW for tasks like language modeling.

Cons:

- Limited Context Window: Can't capture long-range dependencies (e.g., beyond 3-4 words).
- Data Sparsity: Rare n-grams can lead to poor generalization.
- Scalability Issues: Large n-grams increase computational and memory demands.



P("There was heavy rain") ~ P("There") P("was"|"'There") P("heavy"|"was") P("rain"|"heavy")

Language Models: From Basics to Modern Approaches

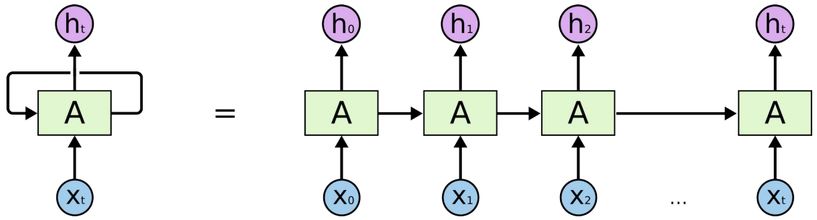
Recurrent Neural Networks

Advantages of RNNs:

- Handle sequential data, including text, speech, and time series.
- Process inputs of any length, unlike feedforward neural networks.
- Share weights across time steps, enhancing training efficiency.

Disadvantages of RNNs:

- Prone to vanishing and exploding gradient problems.
- Training can be challenging, especially for long sequences.
- Computationally slower than other neural network architectures.



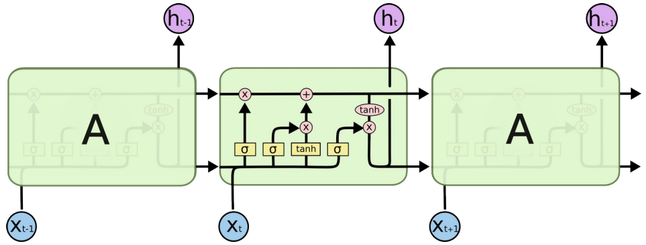
Long-Short Term Memory Networks

Advantages of LSTMs

- Address the Vanishing Gradient Problem
- Capable of Learning Long-Range Dependencies
- Effective Handling of Sequential Data
- Improved Gradient Flow

Disadvantages of LSTMs

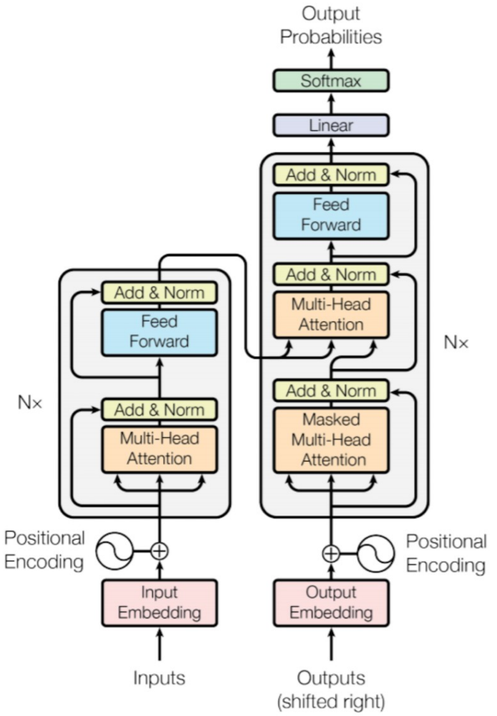
- Computational Complexity
- Memory Usage
- Overfitting on Small Datasets
- Slower Inference and Training



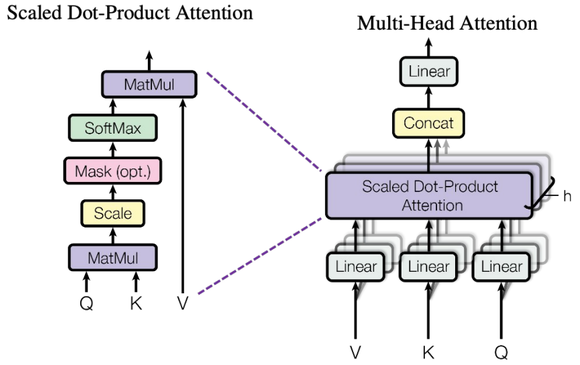
Transformers and the Rise of Modern LMs

Transformers and the Rise of Modern LMs

A neural network architecture designed for handling sequential data using a **self-attention mechanism**. Introduced in the paper *"Attention Is All You Need"* (2017), Transformers are the foundation for modern models like BERT, GPT, and T5.



In the following modules we will delve into the structure and training of transformer-based models.



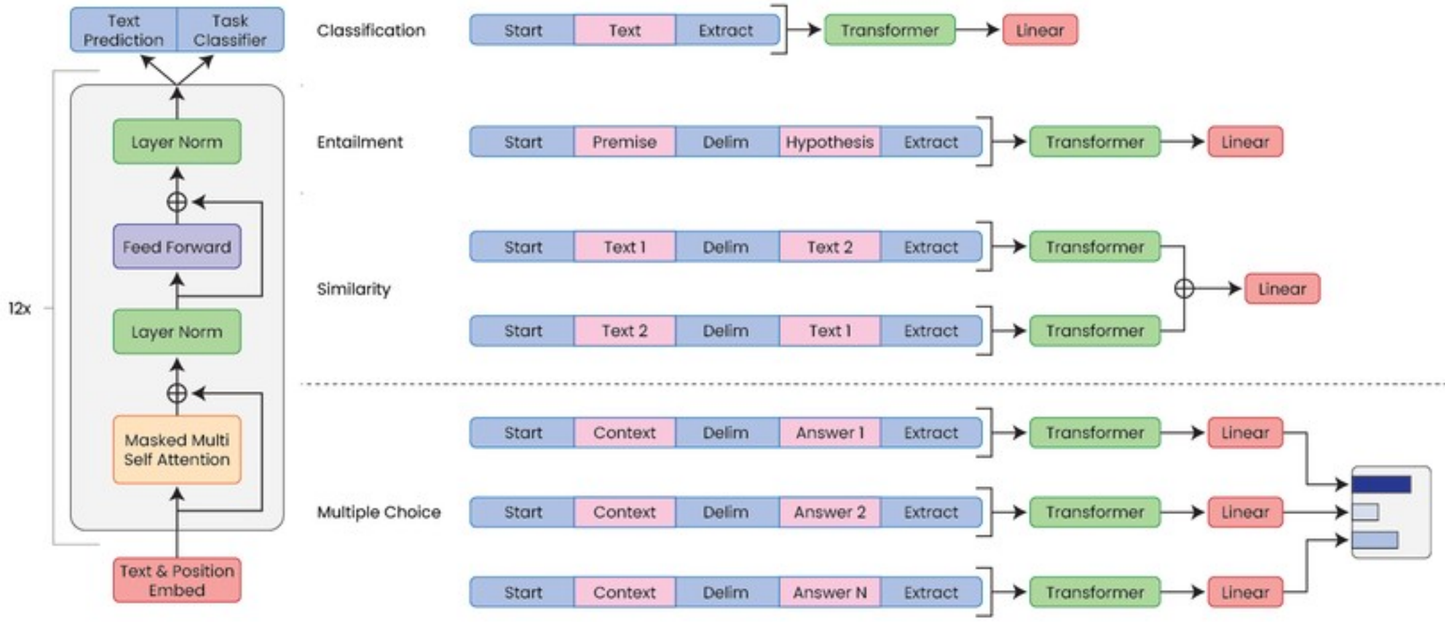
Transformers and the Rise of Modern LMs

Pros:

- Handles Long-Range Dependencies
- Parallelizable
- Scalability
- State-of-the-Art Performance

Cons:

- High Computational Cost
- Data Hungry
- Lack of Recurrence
- Interpretability Issues



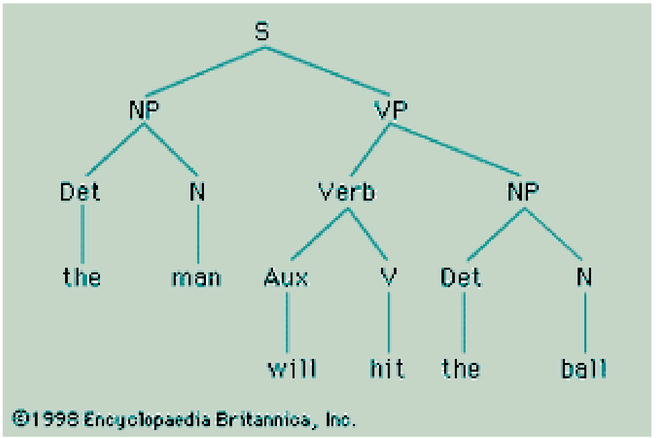
Wrap Up

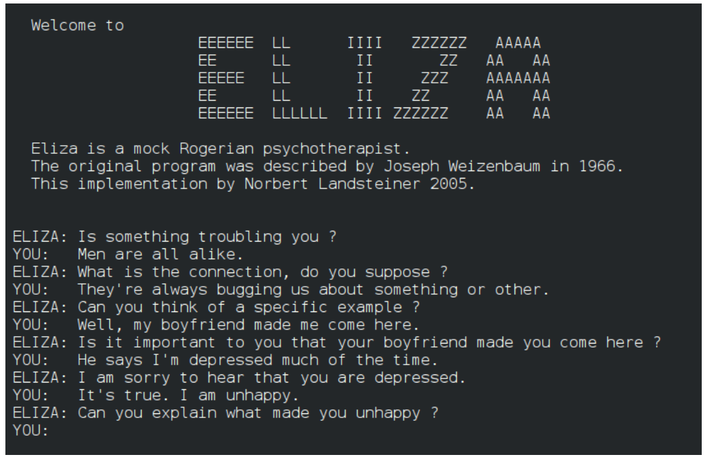
Natural Language Processing and Language Modeling

- Today we introduced natural language processing and language models
- We saw the evolution of NLP approaches from the 1950s rules-based methods to the current deep learning approaches
- Language models were described as a family of probabilistic models used to predict/classify the next word which would solve or naturally follow a given sequence

---

In the next class we will explore tokens and tokenization to see how these models connect language with computing





Thank you!